

REMARKS

Applicants respectfully request reconsideration of this application as amended. By this amendment the Abstract has been amended in accordance with the Examiner's recommendation and the claims amended to remove artifacts of European practice.

Independent Claim 1 includes the feature of coding the reference to a method or an attribute in the linked pseudocode of a module in three bytes constituted by an indicator indicating the reference to a class internal or external to the module, the number of the class, and either the number of the method or the number of the attribute, and loading one or more application program interface modules comprising system classes or service modules, each corresponding to an application, a reference to an external class being systematically interpreted by the virtual machine as a reference to an application program interface module.

Independent Claim 8 recites that the reference to a method or an attribute is coded in at least three bytes corresponding to a reference to a class internal or external to the module, a reference external to the module constituting the index of the API module in the module array, a class number corresponding to the index of the class in the table representing the classes of the module, and a method or attribute number corresponding to the index or the method or the attribute in the array representing the methods or attributes of the class of the module.

Independent Claim 14 recites the reference to a method or an attribute in the pseudocode, is coded in at least three bytes corresponding to a reference to a class internal or external to the module, a class number and a method or attribute number, a reference external to the module is interpreted by the virtual machine as a reference to the index of an API module in the array of the API module or modules.

Independent Claims 1, 8, and 14 clearly include the feature of a reference to a class internal or external to the module and the coding in three bytes.

The Examiner concedes "Wilkinson's does not explicitly disclose reference to an external class as a reference to an API. However, Judge discloses a means for managing embedded systems where an API provides functionality for loading class files (column 6, lines 22-42)." (Reproduced below)

In the preferred embodiment, Application Manager 24 is implemented as a Java network server application, which accepts requests coded into an application protocol 48 such as Java.RTM. Remote Method Invocation (RMI) protocol, HyperText Transport Protocol

(HTTP) for Web hosted requests, or an application level protocol over TCP/IP sockets. As previously described, one type of request in the protocol 48 is the downloading of an application's Java class file 40. In the preferred embodiment, downloading of class files 40 are performed by calling method `loadApplClass()` or method `loadAndInit()`. As previously described, a class file 40 is a binary Java bytecode stream used by the JVM 22 to create a Java class object 30a, 30b, 30c. In addition to downloading the application class file 40 into memory 50, any base classes that the application 26a, 26b, 26c inherits or that interface with the application classes 28a, 28b contained in the class file 40 are also downloaded into an application cache 52 in memory 50. Because the Application Manager 24 allows for caching of class files 40 in memory 50, the application protocol 48 allows for the execution of an already downloaded class file residing in memory 50 via method `initAppl()` followed by a `startAppl()` method call. Once a class file 40 is downloaded, the Java.RTM. `ClassLoader` API is used to instantiate the class objects 28a, 28b associated with the class file(s) 40. The `ClassLoader.defineClass()` method constructs a class object 28a, 28b from the class file byte array. Once defined, the class must be resolved to perform class linking, which allows instance object creation and methods calling. The resolution process is initiated by calling the Java.RTM. `ClassLoader.resolveClass()` API.

Applicants respectfully submit that is readily apparent that not only does Judge fail to teach or suggest an external class as a reference to an API, but moreover none of the cited references teach the coding as claimed. Specifically, none of the cited references teach or suggest the three bytes constituted by an indicator indicating the reference to a class internal or external to the module, the number of the class, and either the number of the method or the number of the attribute as recited in Claim 1, three bytes corresponding to a reference to a class internal or external to the module, a reference external to the module constituting the index of the API module in the module array, a class number corresponding to the index of the class in the table representing the classes of the module, and a method or attribute number corresponding to the index or the method or the attribute in the array representing the methods or attributes of the class of the module as recited in Claim 8, nor three bytes corresponding to a reference to a class internal or external to the module, a class number and a method or attribute number as recited in Claim 14.

In light of these deficiencies the outstanding rejection of the independent claims is untenable and should be withdrawn. Moreover, the dependent claims should also be allowable for at least the above reasons and the additional features recited therein.

Should the Examiner believe that any further action is necessary to place this application in better form for allowance, the Examiner is invited to contact Applicants' representative at the telephone number listed below.

The Commissioner is hereby authorized to charge to Deposit Account No. 50-1165 (T2146-907404) any fees under 37 C.F.R. §§ 1.16 and 1.17 that may be required by this paper and to credit any overpayment to that Account. If any extension of time is required in connection with the filing of this paper and has not been separately requested, such extension is hereby requested.

Respectfully submitted,

Date: April 29, 2005

By: _____

Edward J. Kondracki
Reg. No. 20,604

Jason H. Vick
Reg. No.: 45,285

Miles & Stockbridge, P.C.
1751 Pinnacle Drive
Suite 500
McLean, Virginia 22102-3833
(703) 903-9000